|  |
| --- |
| Taoliji |
| Taolijie |
| Java编码规范 |
|  |
| **王鸿飞** |
| **2014-8-16** |

|  |
| --- |
|  |

# 简介

## 目的

本文档介绍Taolijie服务端Java语言编码规范。

## 预期读者

* 后端开发人员

# 细则

## 代码排版

### 关键字前后必须有空格



if (x == y);

a = b + c;

while (obj.isEmpty()) {

}



if(x==y);

a=b+c;

while(obj.isEmpty()){

}

### 子语句必须放在大括号中



if (x == y) {

foo();

}



if (x == y)

foo();

### 合理使用空行



Hello o = new Hello();

o.sayHello();

Hello o2 = new Hello();

o.sayHi();

Girls g = new Girls();

g.fuck();



Hello o = new Hello();

o.sayHello();

Hello o2 = new Hello();

o.sayHi();

Girls g = new Girls();

g.fuck();

## 小驼峰命名法

所有名称以小写字母开头,

名字可以由多个英语单词组成，每个单词首字母大写。如：

checkYourMoney();

merryYourGirlfriend();

String blueBall = new Ball();

int fatherAge = 45;

int myHeight = 177;

## 类命名

### 所有工具类以\*\*\*Utils结尾：

StringUtils, DateUtils, LogUtils

### 首字母大写

class JobController;

class AccountService;

### 抽象类以Abstract\*\*开头

abstract class AbstractContainer;

abstract class AbstractList;

### Service层的类以\*\*Service结尾

MailService, AccountService, CategoryService

### DAO层的类以insert\*\*, update\*\*, select\*\*, find\*\*, search\*\*开头

insertUser(); updateUserInfo(); selectUserById();

## 方法命名

以最能解释说明该方法的功能为原则。如，

updateUserInfo(); //修改用户信息

changePassword(); // 修改密码

loginByUsername(); // 使用用户名登陆

getOnlineUserAmount(); // 得到在线用户数

**严禁以数字结尾**foo1(), foo2(), foo3()命名。

## 变量命名

以最能解释说明变量的用途为原则，如：

String userAddr; // 用户地址

Date nowTime; // 现在的系统日期

Double mathScores; // 数学成绩

常量以大写命名，如：

DEFAULT\_PORT = 80;

LOGIN\_URL = “/login”;

**严禁出现a, b, c, i, j, x, y这样无意义的变量名**, for循环变量除外。

## 单词缩写

为了减少敲击键盘的次数，可以使用一些众所周知的(well-known)缩写：

address 🡪 addr

information 🡪 info

password 🡪 pwd

string 🡪 str

object 🡪 obj

等等

# Git代码提交

commit时必须写一段简短的文字，说明这次提交改了哪些内容，或者添加、修改了哪些功能。如：

git commit –m “修复二手页面500错误”

git commit –m “重构控制器代码”

git commit –m “添加私信功能”

commit提交的原则是尽量多提交，每次提交一小部分改动，不要等积攒了一大堆代码后再提交。